

03-22-00

A

Please type a plus sign (+) inside this box → ☒

PTO/SB/05 (4/98)  
 Approved for use through 09/30/2000. OMB 0651-0032  
 Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE  
 Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

<b>UTILITY PATENT APPLICATION TRANSMITTAL</b> <small>(Only for new nonprovisional applications under 37 C.F.R. § 1.53(b))</small>	Attorney Docket No.	
	First Inventor or Application Identifier	Zhe Li
	Title	Method for Conditional Tautology Checking
	Express Mail Label No.	EJ418535055US

<b>APPLICATION ELEMENTS</b> <small>See MPEP chapter 600 concerning utility patent application contents.</small>	<b>ADDRESS TO:</b> Assistant Commissioner for Patents Box Patent Application Washington, DC 20231
1. <input checked="" type="checkbox"/> * Fee Transmittal Form (e.g., PTO/SB/17) <small>(Submit an original and a duplicate for fee processing)</small> 2. <input checked="" type="checkbox"/> Specification [Total Pages <b>22</b> ] <small>(preferred arrangement set forth below)</small> - Descriptive title of the invention - Cross References to Related Applications - Statement Regarding Fed sponsored R & D - Reference to Microfiche Appendix - Background of the invention - Brief Summary of the invention - Brief Description of the Drawings (if filed) - Detailed Description - Claim(s) - Abstract of the Disclosure 3. <input checked="" type="checkbox"/> Drawing(s) (35 U.S.C. 113) [Total Sheets <b>3</b> ] 4. Oath or Declaration [Total Pages <b>2</b> ] a. <input checked="" type="checkbox"/> Newly executed (original or copy) b. <input type="checkbox"/> Copy from a prior application (37 C.F.R. § 1.63(d)) <small>(for continuation/divisional with Box 16 completed)</small> i. <input type="checkbox"/> DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 C.F.R. §§ 1.63(d)(2) and 1.33(b).	5. <input type="checkbox"/> Microfiche Computer Program (Appendix) 6. Nucleotide and/or Amino Acid Sequence Submission <small>(if applicable, all necessary)</small> a. <input type="checkbox"/> Computer Readable Copy b. <input type="checkbox"/> Paper Copy (identical to computer copy) c. <input type="checkbox"/> Statement verifying identity of above copies
<b>ACCOMPANYING APPLICATION PARTS</b> 7. <input type="checkbox"/> Assignment Papers (cover sheet & document(s)) 8. <input type="checkbox"/> 37 C.F.R. § 3.73(b) Statement <input type="checkbox"/> Power of Attorney <small>(when there is an assignee)</small> 9. <input type="checkbox"/> English Translation Document (if applicable) 10. <input type="checkbox"/> Information Disclosure Statement (IDS)/PTO-1449 <input type="checkbox"/> Copies of IDS Citations 11. <input type="checkbox"/> Preliminary Amendment 12. <input checked="" type="checkbox"/> Return Receipt Postcard (MPEP 503) <small>(Should be specifically itemized)</small> 13. <input checked="" type="checkbox"/> * Small Entity Statement(s) <input type="checkbox"/> Statement filed in prior application, Status still proper and desired <small>(PTO/SB/09-12)</small> 14. <input type="checkbox"/> Certified Copy of Priority Document(s) <small>(if foreign priority is claimed)</small> 15. <input type="checkbox"/> Other:	
<b>NOTE FOR ITEMS 1 &amp; 13: IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.27), EXCEPT IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28).</b>	

16. If a **CONTINUING APPLICATION**, check appropriate box, and supply the requisite information below and in a preliminary amendment:  
☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No. \_\_\_\_\_  
 Prior application information: Examiner \_\_\_\_\_ Group / Art Unit: \_\_\_\_\_  
**For CONTINUATION or DIVISIONAL APPS only:** The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

17. **CORRESPONDENCE ADDRESS**

☐ Customer Number or Bar Code Label (Insert Customer No. or Attach bar code label here) or ☒ Correspondence address below

Name	Zhe Li				
Address	1 Argent Drive				
City	Poughkeepsie	State	New York	Zip Code	12603
Country	USA	Telephone	914-298-8342	Fax	914-298-1889

Name (Print/Type)	Zhe Li	Registration No. (Attorney/Agent)	
Signature	<i>Zhe Li</i>	Date	03/21/2000

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

03/21/00  
 Jc600 U.S. PTO

00534333 03/21/00

JC564 U.S. PTO  
 09/531633  
 03/21/00

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

**STATEMENT CLAIMING SMALL ENTITY STATUS  
(37 CFR 1.9(f) & 1.27(b))--INDEPENDENT INVENTOR**

Docket Number (Optional)

Applicant, Patentee, or Identifier: Zhe Li

Application or Patent No.: \_\_\_\_\_

Filed or Issued: \_\_\_\_\_

Title: Method for Conditional Tautology Checking

As a below named inventor, I hereby state that I qualify as an independent inventor as defined in 37 CFR 1.9(c) for purposes of paying reduced fees to the Patent and Trademark Office described in:

- ☒ the specification filed herewith with title as listed above.  
☐ the application identified above.  
☐ the patent identified above.

I have not assigned, granted, conveyed, or licensed, and am under no obligation under contract or law to assign, grant, convey, or license, any rights in the invention to any person who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person had made the invention, or to any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern, or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey, or license any rights in the invention is listed below:

- ☒ No such person, concern, or organization exists.  
☐ Each such person, concern, or organization is listed below.

Separate statements are required from each named person, concern, or organization having rights to the invention stating their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

Zhe Li

NAME OF INVENTOR

NAME OF INVENTOR

NAME OF INVENTOR

Zhe Li

Signature of inventor

Signature of inventor

Signature of inventor

03 / 21 / 2000

Date

Date

Date

Patent Application of

Zhe Li

For

**METHOD FOR CONDITIONAL TAUTOLOGY CHECKING**

**CROSS-REFERENCE TO RELATED APPLICATIONS**

This application claims priority to provisional application No. 60/125835, titled Partial Tautology Checking Method Using Partial Enumeration and Simplification, and filed on March 24<sup>th</sup>, 1999, the contents of which are herein incorporated by reference..

**STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR  
DEVELOPMENT**

Not Applicable.

**REFERENCE TO A MICROFICHE APPENDIX**

Not Applicable.

## BACKGROUND OF THE INVENTION

This invention relates to Boolean functions, specifically to checking whether a Boolean function is a tautology with a given constraint.

The value of a Boolean function can be either 0 or 1. A Boolean function can depend on any number of variables, and each variable's value can be either 0 or 1. The Boolean function maps each combination of values of these variables to either 0 or 1. A Boolean function is satisfiable if it maps at least one such combination to 1. A Boolean function is a tautology if it maps all combinations to 1. A Boolean function is a tautology if and only if the negation of the Boolean function is not satisfiable.

A tautology checking method decides whether a given Boolean function is a tautology. An equivalence checking method decides whether two given Boolean functions are equivalent. An equivalence checking method can be used as a tautology checking method if one of the two given Boolean functions is Boolean constant 1. A tautology checking method can perform equivalence checking if the given Boolean function is the exclusive nor (XNOR) of two Boolean functions.

Because the value of a signal in a digital circuit can be either 1 or 0, digital circuits are often represented as Boolean functions. Equivalence checking methods and tautology checking methods can be used to check whether two digital circuits are equivalent, which is used in functional design verification of digital circuits. The verification is generally performed as checking the equivalence between a

specification and an implementation. These methods can be very useful when using computers to execute them.

As digital circuits become more and more complex while integrated circuit technologies grow, equivalence checking methods and tautology checking methods often fail to handle the complex Boolean functions used to represent digital circuits. The failures are either due to unreasonably long run times or due to requiring unreasonably large amounts of computer memory.

A way to avoid such failures is to check equivalence or tautology for only some of all combinations of the variable values. This is used widely in the industry because each of the ignored combinations is often similar enough to a combination involved in the checking or some combinations are less critical than others. This is normally performed as logic simulation for one combination at a time. When the same combinations are involved, logic simulation is usually much less efficient than equivalence checking methods or tautology checking methods.

There clearly are needs of more efficient methods for checking equivalence or tautology for a given subset of all combinations of the variable values.

## **BRIEF SUMMARY OF THE INVENTION**

The present invention provides a method for determining whether a Boolean function is equivalent to Boolean constant 1 within a given subset of the input space.

The given subset is divided into a plurality of smaller subsets regardless how the smaller subsets are chosen. If any of the smaller subsets is not a cube, this smaller subset is divided further. If one of the smaller subsets is a cube, the Boolean function is simplified with constant substitution within the cube. If the simplification result is not a Boolean constant, the cube is divided further. If the simplification result is Boolean constant 0, a negative conclusion is reached. The conclusion is positive if none of the simplification result is Boolean constant 0.

Many of the subset division steps and many of the Boolean function simplification steps can be performed independently of one another, and therefore these independent operations can be performed separately at different times or on different computers. The given subset can expand or shrink dynamically if updates of the given subset are considered in the steps. These dynamic updates make it possible to dynamically adjust the divided subsets.

## BRIEF DESCRIPTION OF DRAWINGS

FIG. 1 illustrates, in flow diagram form, a recursive process for conditional tautology checking.

FIG. 2 illustrates, in flow diagram form, an iterative process for conditional tautology checking.

FIG. 3 is a representative computer system in conjunction with which the embodiments of the present invention may be implemented.

## DETAILED DESCRIPTION OF THE INVENTION

A method for automatically checking whether a Boolean function is equivalent to Boolean constant 1 with a given constraint, using a computer, is disclosed.

A Boolean function is a tautology if the Boolean function maps all points in the entire input space to Boolean constant 1. A tautology is equivalent to Boolean constant 1.

A Boolean function is a conditional tautology if the Boolean function maps all points in a given subset of the input space to Boolean constant 1, while it may map any point outside the given subset of the input space to either Boolean constant. A tautology is a conditional tautology when the entire input space is its given subset of the input space.

Conditional tautologies are easy to check because of the flexibility of doing divide-and-conquer with them. A Boolean function is a conditional tautology with a given subset of the input space if

- (1) a number of smaller conditional tautologies all share this Boolean function, and
- (2) these smaller conditional tautologies' given subsets of the input space jointly cover exactly the large conditional tautology's given subset of the input space

(i.e. the large conditional tautology's given subset of the input space is the union of these smaller conditional tautology's given subsets of the input space).

A cube is a subset of the input space where some input variables are substituted with Boolean constant 1 and some other input variables are substituted with Boolean constant 0. The count of points in a cube must be 1, 2, 4, or any other power of 2 because zero or more input variables are completely free to take any values. A Boolean function is a conditional tautology with a cube as the given subset of the input space if the Boolean function is simplified to Boolean constant 1 after the input variables in the Boolean function are substituted with these 1's or 0's according to the cube's definition.

Conditional tautology checking can be performed in a recursive process as illustrated in FIG. 1. The process starts with a step 100 receiving a Boolean function and a given subset of the input space. A step 110 is performed next to determine whether to go to a step 120 or a step 150. It must go to step 150 if the given subset of the input space is not a cube. It may go to either step 120 or step 150 under other conditions, but it must go to step 120 if the given subset of the input space includes only one point. The Boolean function is simplified in step 120 with substituting input variables in the Boolean function with Boolean constants according to the cube's definition. A step 130 is performed after step 120 to check the simplification result before going to step 150 or another step 140. Step 150 is performed next if the simplification result is not a Boolean constant. Otherwise step 140 is performed next

to look at the Boolean constant resulted from the simplification. The positive conclusion is given in a step 180 after step 140 if the simplification result is Boolean constant 1. The negative conclusion is given in a step 190 after step 140 if the simplification result is Boolean constant 0. The negative conclusion may include at least one counter example indicating the substitution of the input variables with the Boolean constants.

In step 150, the given subset of the input space is divided into several smaller subsets of the input space before moving on to a step 160. For each of these smaller subsets of the input space, the conditional tautology checking process, starting from step 100, is performed recursively for the Boolean function (the simplified copy if after step 120) with this smaller subset as the given subset of input space. After all such recursions of step 160, a step 170 is performed to summarize all conclusions collected from these recursions. If all the collected conclusions are positive, the positive conclusion is given in step 180. If any of the collected conclusions is negative, the negative conclusion is given in step 190.

The strategy of dividing a subset of the input space into smaller ones, in step 150, can be arbitrary (manual or automatic), and a process of conditional tautology checking can use a mixture of several such strategies. Any strategy can get correct answer after a finite number of the recursive divisions because all smaller subsets of the input space cannot get smaller than the smallest cubes (also known as individual points of the input space). Because each of the smallest cubes has all input

variables substituted with Boolean constants, it always gets the Boolean function simplified to a Boolean constant after a finite number of divisions.

In some cases, the strategy of dividing a subset of the input space, in step 150, is to divide it into the largest cubes within the subset because this causes applying simplification, in step 120, as early as possible. In some other cases, smaller cubes are preferred because large cubes have more chances to waste the simplification efforts in step 120. A larger cube often has smaller chance to result in a Boolean constant from simplification with constant substitution in step 120. These cubes may overlap with each other, and it depends on specific cases whether it is better to allow this overlapping or not.

Conditional tautology checking can also be performed in the following iterative process illustrated in FIG. 2. The iterative process starts with a step 200 receiving a Boolean function and a given subset of the input space. A step 210 is performed next to divide the given subset of the input space into two subsets. One is called a cube subset and the other is called a tail subset. The cube subset is a cube in the input space. If the given subset is a cube itself and it is chosen to be the cube subset, the tail subset is empty. A step 220 is performed after step 210 to simplify the Boolean function with substituting input variables in the Boolean function with Boolean constants according to the cube's definition. The simplification result can be used to replace the original Boolean function in all following steps and iterations if the tail subset is empty. Otherwise the original Boolean function is preserved for all

following steps and iterations, and the simplification result can be used temporarily as the Boolean function to simplify again in the next round, as described below, only when the simplification result is not a Boolean constant.

A step 230 is then performed to check the simplification result before performing a step 240 or another step 250. If the simplification result is not a Boolean constant, step 240 is performed to make the cube subset a smaller cube by dividing the given subset of input space similarly to in step 210, and step 220 is performed again after step 240 for the smaller cube with the last simplification result as the Boolean function to simplify. Otherwise step 250 is performed next to look at the Boolean constant resulted from the simplification before performing a step 260 or another step 270. The negative conclusion is given in step 260 if the simplification result is Boolean constant 0. The negative conclusion may include at least one counter example indicating the substitution of the input variables with the Boolean constants. If the simplification result is Boolean constant 1, step 270 is performed to examine whether the tail subset from step 210 or step 240 is empty. If the tail subset is empty, the positive conclusion is given in a step 290. Otherwise a step 280 is performed making the tail subset the new given subset of the input space before looping back to step 210.

There are many ways to represent a Boolean function in a computer for a program to process. This invention is not limited to any particular representation because the representation of Boolean functions is only involved in the simplification

and the constant substitution within steps 120 and 220.

There are different ways to represent a subset of the input space. One way is to use a characteristic function. A Boolean function is a characteristic function of a subset of the input space if its value is 1 for and only for all points in the subset. Another way is to use ranges of binary integers.

Given an ordering of all the input variables, each point of the input space can be represented as a binary integer. Each bit of the binary integer represents the value of each input variable. The total count of these binary integers is the  $N$ th power of two where  $N$  is the number of input variables. The lower bound of these binary integers is zero. A range of these binary integers represents a subset of the input space. A range has a lower bound and an upper bound, and it includes all binary integers between the two bounds (i.e. all binary integers that are greater than or equal to the lower bound and smaller than or equal to the upper bound). Any subset of the input space can be represented as one or more ranges of binary integers jointly.

A range of these binary integers represents a cube if the only difference between the range's upper bound and the range's lower bound is at the rightmost bits (i.e. the rightmost bits of the range's lower bound, if they are 0's, all correspond to 1's in the rightmost bits of the range's upper bound, respectively). A range from

10100 to 10101 represents a cube. A range from 10100 to 10110 does not represent a cube.

Given any range, the range representing a cube that covers the first part of the given range can be easily identified because

- (1) it shares the lower bound with the given range,
- (2) its upper bound is obtained by turning the rightmost zero or more 0's of its lower bound into 1's, and
- (3) its upper bound is smaller than or equal to the given range's upper bound.

If several cubes satisfy these three conditions, one cube is selected depending on the chosen strategy. The selected cube divides the given range into two ranges: a range representing the cube and the other range representing the rest.

Based on the range representation of subsets of the input space, the conditional tautology checking method can be modified to handle a range whose upper bound is dynamically shifting. This shifting is useful because it enables dynamic division of a subset of the input space into several. The division criteria can be dynamically decided on-the-fly based on the changing factors of the computing environment. These changing factors can include any interrupts, memory shortage and time limit. It is the most interesting if the range is shrinking so that the cube used for simplification at the moment is completely outside the range: simply go to step

290 if it is in any other step. If the range is extending, the tail subset need to be extended to the upper bound of the range before step 270.

Furthermore, checking different conditional tautologies can be in different computing environment. For example, some smaller conditional tautologies can be moved to other computers to continue checking.

More generally, checking a conditional tautology can be performed as checking several smaller conditional tautologies concurrently if the division is decided in advance. Then each of the smaller conditional tautologies can be checked with dynamic divisions. Checking each conditional tautology can start from either end of the range representing the given subset of the input space. It can start from different ends for checking different conditional tautologies. If desired, the iterative process and the recursive process can be mixed in checking a conditional tautology. Different processes can be used either at different levels of the divisions or for different smaller subsets of the input space from the same division. When the (given or smaller) subset of the input space is a cube, any other tautology checking method other than simplification can also be applied after constant substitution.

With these flexible division techniques, very large Boolean functions can be handled in tautology checking. This is required for automatically verifying large data-processing systems using computers, especially when verifying them against the expected behaviors without any assumptions of their internal structures.

The invention discussed above may be implemented within dedicated hardware 15 as illustrated in FIG. 3 or within processes implemented within a data processing system 13. A typical hardware configuration of a workstation, that may be implemented to accomplish the method disclosed herein, is illustrated and includes a central processing unit (CPU) 10, such as a conventional microprocessor, and a number of other units interconnected via a system bus 12. The workstation shown in FIG. 3 includes random access memory (RAM) 14, read only memory (ROM) 16, and input/output (I/O) adapter 18 for connecting peripheral devices, such as disk units 20 and tape units 40, to bus 12. A user interface adapter 22 is used to connect a keyboard device 24 and a mouse 26 to system bus 12. Other user interface devices such as a touch screen device (not shown) may also be coupled to system bus 12 through user interface adapter 22.

A communication adapter 34 is also shown for connecting the workstation to a data processing network 17. Further, a display adapter 36 connects system bus 12 to a display device 38. The method of the present invention may be implemented and stored in one or more of disk units 20, tape drive 40, ROM 16 and/or RAM 14, or even made available to system 13 via network 17 through communication adapter 34 and thereafter processed by CPU 10. Since the apparatus implementing the present invention is, for the most part, composed of electronic components and circuits known to those skilled in the art, circuit details will not be explained in any greater extent than that considered necessary as illustrated above, for the

understanding and appreciation of the underlying concepts of the present invention and in order not to obfuscate or distract from the teachings of the present invention.

While the above invention has been described with reference to certain preferred embodiments, the scope of the present invention is not limited to these embodiments. One skilled in the art may find variations of these preferred embodiments that, nevertheless, fall within the spirit of the present invention, whose scope is defined by the claims set forth below.

## CLAIMS

I claim:

1. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, comprising the steps of:

receiving said Boolean function, said Boolean constant and said given subset of the input space;

dividing said given subset of the input space into a set of a plurality of smaller subsets of the input space, whereby the conclusion is positive if said Boolean function is equivalent to said Boolean constant within every member of said set of a plurality of smaller subsets of the input space.

2. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 1 further comprising the step of:

determining whether said given subset of the input space is a cube; and

simplifying, if said given subset of the input space is a cube, said Boolean function with substituting the input variables in said Boolean function with Boolean constants according to the definition of said cube, whereby the conclusion is positive if the simplification result is said Boolean constant, and the conclusion is negative if the simplification result is a Boolean constant other than said Boolean constant.

3. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 2 further comprising the step of:

replacing said Boolean function with the simplification result of the simplifying step before the dividing step, whereby the simplification result is used as said Boolean function in all later steps.

4. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 1 wherein a member of said set of a plurality of smaller subsets of the input space is a cube within said given subset of the input space, whereby it is possible to substitute the input variables in said Boolean function with Boolean constants according to the definition of said cube.

5. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 4 further comprising the step of:

simplifying said Boolean function with substituting the input variables in said Boolean function with Boolean constants according to the definition of said cube, whereby the conclusion is positive if the simplification result is said Boolean constant, and the conclusion is negative if the simplification result is the Boolean constant other than said Boolean constant.

6. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 5 further comprising the step of:

repeating the dividing step and the simplifying step if the simplification result is not a Boolean constant, whereby said cube is smaller and the simplification result will eventually be a Boolean constant when said cube is small enough.

7. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 4 wherein said given subset of the input space is represented as a first range of binary integers.

8. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 7 wherein said set of a plurality of smaller subsets of the input space has only two members: said cube which is represented as a second range of binary integers, and a second member which is represented as a third range of binary integers, whereby said third range of binary integers is divided repeatedly into cubes and such divisions do not need to all complete if the negative conclusion is given for any of these cubes.

9. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 8 further comprising the step of:

replacing said given subset of the input space, represented by said first range of binary integers, with said second member of said set of a plurality of smaller subsets of the input space, represented by said third range of binary integers, after finishing all steps related to said cube and said second range of binary integers which represents said cube, whereby said given subset of the input space become smaller and smaller and whether said Boolean function is equivalent to said Boolean constant within said given subset of the input space can be determined using simplification when said given subset of the input space eventually becomes a cube.

10. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 9 further comprising the step of:

shifting a boundary of said third range of binary integers before the replacing step if the corresponding boundary of said first range of binary integers is shifted, whereby a boundary of said first range of binary integers can shift dynamically.

11. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 4 further comprising the step of:

substituting the input variables in said Boolean function with Boolean constants according to the definition of said cube, whereby any tautology checking method can be used to process the Boolean function resulted from the substitution.

12. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 11 wherein said given subset of the input space is represented as a first range of binary integers.

13. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 12 wherein said set of a plurality of smaller subsets of the input space has only two members: said cube which is represented as a second range of binary integers, and a second member which is represented as a third range of binary integers, whereby said third range of binary integers is divided repeatedly into cubes and such divisions do not need to all complete if the negative conclusion is given for any of these cubes.

14. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 13 further comprising the step of:

replacing said given subset of the input space, represented by said first range of binary integers, with said second member of said set of a plurality of smaller subsets of the input space, represented by said third range of binary integers, after finishing all steps related to said cube and said second range of binary integers which represents said cube, whereby said given subset of the input space become smaller and smaller and whether said Boolean function is equivalent to said Boolean

constant within said given subset of the input space can be determined using simplification when said given subset of the input space eventually becomes a cube.

15. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 14 further comprising the step of:

shifting a boundary of said third range of binary integers before the replacing step if the corresponding boundary of said first range of binary integers is shifted, whereby a boundary of said first range of binary integers can shift dynamically.

16. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 1 further comprising the step of:

determining whether said given subset of the input space is a cube; and substituting, if said given subset of the input space is a cube, the input variables in said Boolean function with Boolean constants according to the definition of said cube, whereby any tautology checking method can be used to process the Boolean function resulted from the substitution.

17. A method for determining whether a Boolean function is equivalent to a Boolean constant within a given subset of the input space, as in claim 1 further comprising the step of:

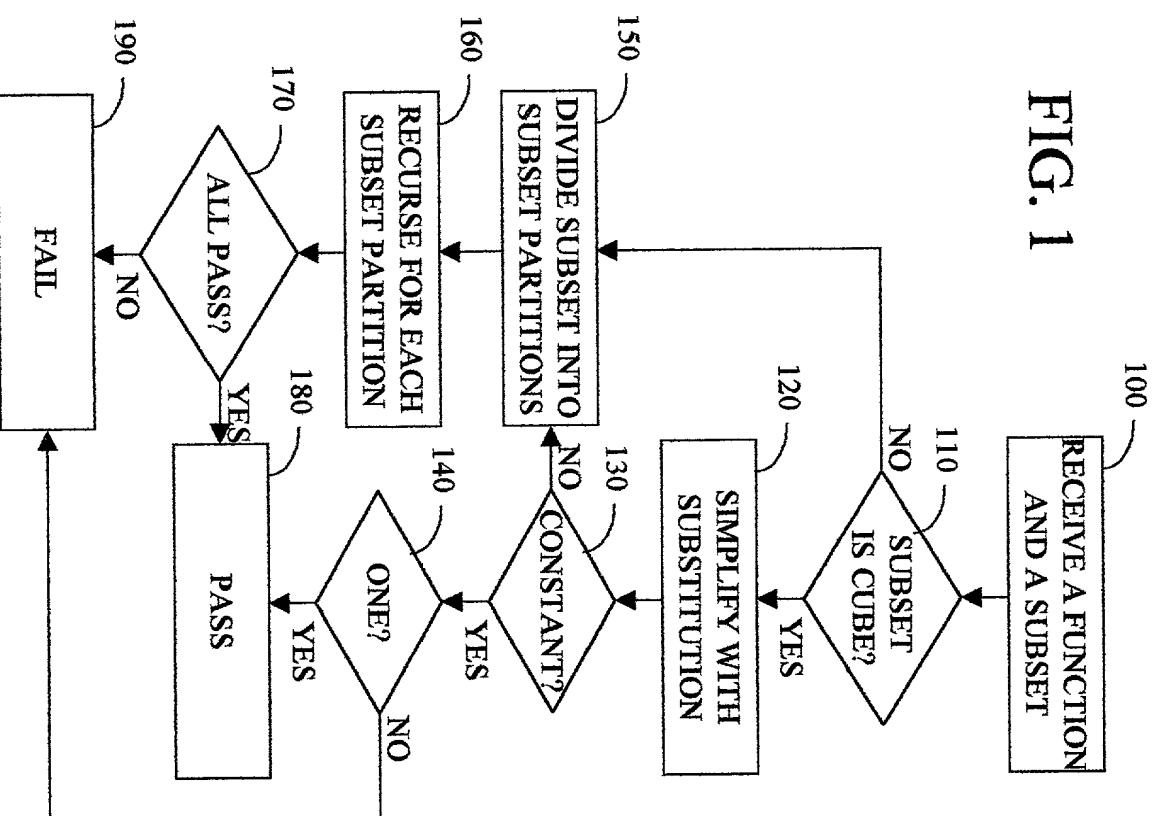
starting a process for each member of said set of a plurality of smaller subsets of the input space determining whether said Boolean function is equivalent to said Boolean constant within said member of said set of a plurality of smaller subsets of the input space, whereby these processes can run on the same computer or on several computers, at the same time or at different times.

## ABSTRACT OF THE DISCLOSURE

The present invention provides a method for determining whether a Boolean function is equivalent to Boolean constant 1 within a given subset of the input space. The given subset is divided into a plurality of smaller subsets regardless how the smaller subsets are chosen. If any of the smaller subsets is not a cube, this smaller subset is divided further. If one of the smaller subsets is a cube, the Boolean function is simplified with constant substitution within the cube. If the simplification result is not a constant, the cube is divided further. If the simplification result is constant 0, a negative conclusion is reached. The conclusion is positive if none of the simplification result is Boolean constant 0.

Many of the subset division steps and many of the Boolean function simplification steps can be performed independently of one another, and therefore these independent operations can be performed separately at different times or on different computers. The given subset can expand or shrink dynamically if updates of the given subset are considered in the steps. These dynamic updates make it possible to dynamically adjust the divided subsets.

FIG. 1



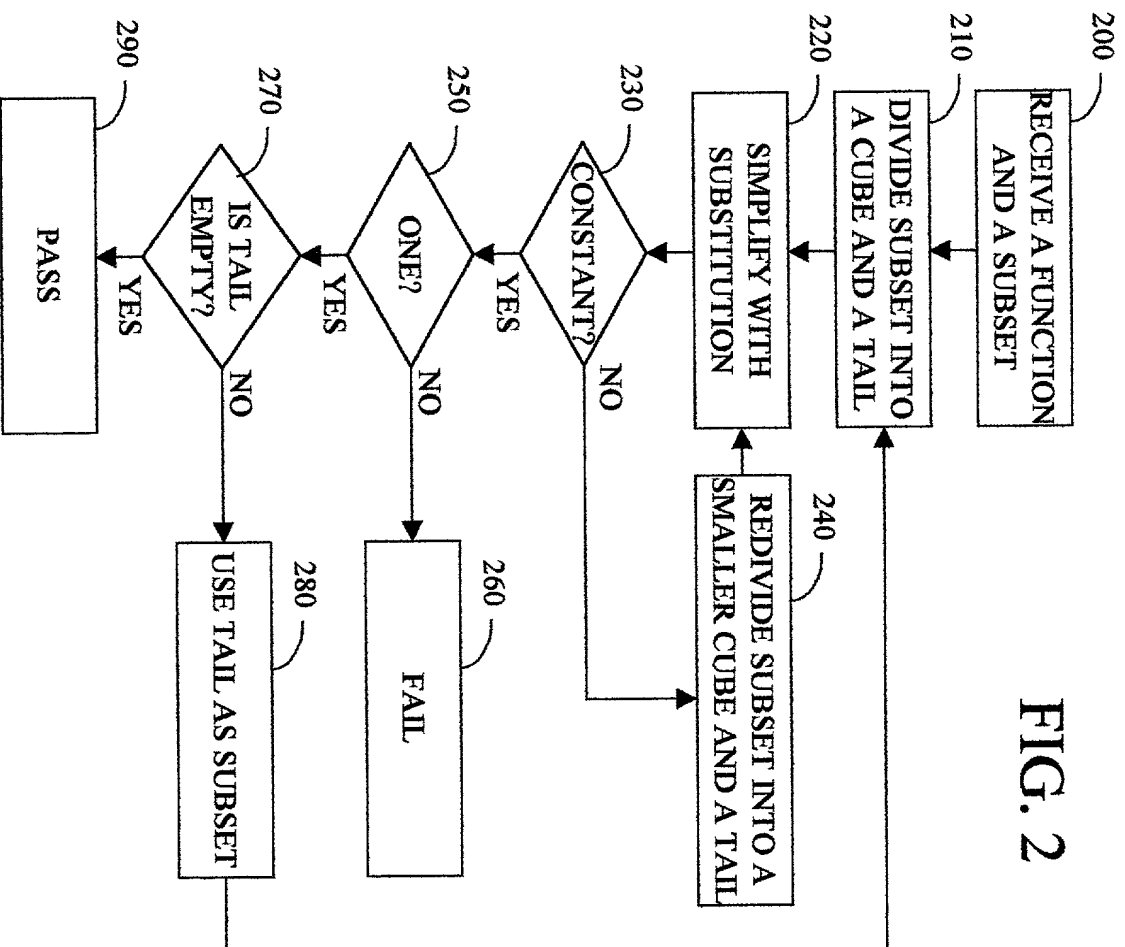


FIG. 2



Please type a plus sign (+) inside this box → ☐

PTO/SB/01 (12-97)

Approved for use through 9/30/00. OMB 0651-0032

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

<b>DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION</b> (37 CFR 1.63)	<b>Attorney Docket Number</b>	
	<b>First Named Inventor</b>	Zhe Li
	<b>COMPLETE IF KNOWN</b>	
	<b>Application Number</b>	/
	<b>Filing Date</b>	
	<b>Group Art Unit</b>	
<input checked="" type="checkbox"/> Declaration Submitted with Initial Filing	OR	<input type="checkbox"/> Declaration Submitted after Initial Filing (surcharge (37 CFR 1.16 (e)) required)
<b>Examiner Name</b>		

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

Method for Conditional Tautology Checking

the specification of which  
☒ is attached hereto  
OR  
☐ was filed on (MM/DD/YYYY) [ ] as United States Application Number or PCT International Application Number [ ] and was amended on (MM/DD/YYYY) [ ] (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment specifically referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. 119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or of any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached?	
				YES	NO
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

☐ Additional foreign application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto:

I hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below.

Application Number(s)	Filing Date (MM/DD/YYYY)	<input type="checkbox"/> Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.
60/125,835	03/24/99	

[Page 1 of 2]

Burden Hour Statement: This form is estimated to take 0.4 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

Please type a plus sign (+) inside this box →



PTO/SB/01 (12-97)

Approved for use through 9/30/00. OMB 0651-0032

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

## DECLARATION — Utility or Design Patent Application

I hereby claim the benefit under 35 U.S.C. 120 of any United States application(s), or 365(c) of any PCT International application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. 112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. Parent Application or PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number (if applicable)

☐ Additional U.S. or PCT international application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.

As a named inventor, I hereby appoint the following registered practitioner(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

☐ Customer Number

OR

☐ Registered practitioner(s) name/registration number listed below

Place Customer Number Bar Code Label here

Name	Registration Number	Name	Registration Number

☐ Additional registered practitioner(s) named on supplemental Registered Practitioner Information sheet PTO/SB/02C attached hereto.

Direct all correspondence to: ☐ Customer Number or Bar Code Label

OR ☒ Correspondence address below

Name	Zhe Li				
Address					
Address	1 Argent Drive				
City	Poughkeepsie	State	NY	ZIP	12603
Country	USA	Telephone	914-298-8342	Fax	914-298-1889

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Name of Sole or First Inventor:

☐ A petition has been filed for this unsigned inventor

Given Name (first and middle [if any])		Family Name or Surname	
Zhe		Li	
Inventor's Signature	Zhe Li		Date 3/21/00
Residence: City	Poughkeepsie	State	NY
Country	USA	Citizenship	China
Post Office Address			
Post Office Address	1 Argent Drive		
City	Poughkeepsie	State	NY
ZIP	12603	Country	USA

☐ Additional inventors are being named on the supplemental Additional Inventor(s) sheet(s) PTO/SB/02A attached hereto